1. Injection

- **Nedir:** Kullanıcının girdiği veriler üzerinden bir uygulamaya kötü niyetli komutların sızdırılması durumudur. Bu komutlar veri tabanına, sistem komutlarına veya bir uygulamanın diğer parçalarına enjekte edilebilir.
- **Neden Kaynaklanır:** Kullanıcı girdilerinin yeterince doğrulanmaması veya temizlenmemesi.
- **Türleri:** SQL Injection, Command Injection gibi türleri vardır.
- **Nasıl Önlenir:** Kullanıcı girdilerini dikkatlice doğrulamak ve parametreli sorgular kullanmak bu sorunun çözümünde etkili olabilir.

2. Broken Authentication

- **Nedir:** Kimlik doğrulama sistemlerinde meydana gelen açıklar, kullanıcı hesaplarının ele geçirilmesine neden olabilir. Parola tahmini, oturum çalma gibi saldırılar bu zafiyetin bir parçasıdır.
- **Neden Kaynaklanır:** Zayıf parola politikaları, oturum yönetiminde yapılan hatalar.
- **Nasıl Önlenir:** Güçlü parolalar, çok faktörlü kimlik doğrulama (MFA) ve oturum sürelerini kısıtlamak.

3. Sensitive Data Exposure

- **Nedir:** Hassas verilerin (şifreler, kredi kartı bilgileri vb.) güvensiz bir şekilde depolanması veya iletilmesi.
- **Neden Kaynaklanır:** Verilerin şifrelenmeden depolanması veya iletilmesi.
- **Nasıl Önlenir:** Güçlü şifreleme algoritmaları kullanmak ve verileri yalnızca güvenli kanallardan iletmek.

4. XML External Entities (XXE)

- **Nedir:** Dış kaynaklı XML verilerinin kötüye kullanılarak sistemdeki dosyalara erişim sağlanması.
- **Neden Kaynaklanır:** Güvenli olmayan XML işleme yapılandırmaları.
- **Nasıl Önlenir:** Dış varlık (external entity) işlemesini devre dışı bırakmak ve güvenli XML işlemleri kullanmak.

5. Broken Access Control

- **Nedir:** Kullanıcıların, yetkileri dışında işlemler yapabilmesi veya bilgilere erişebilmesi durumu.
- **Neden Kaynaklanır:** Erişim kontrollerinin yanlış veya eksik yapılandırılması.
- **Nasıl Önlenir:** Erişim kontrol sistemlerini dikkatlice gözden geçirmek ve doğrulama süreçlerini güçlendirmek.

6. Security Misconfiguration

- **Nedir:** Güvenlik yapılandırmalarının eksik ya da yanlış yapılması.
- **Neden Kaynaklanır:** Varsayılan ayarların kullanılması veya güvenlik yamalarının zamanında uygulanmaması.
- **Nasıl Önlenir:** Güvenlik yapılandırmalarını düzenli olarak gözden geçirmek ve güncellemek.

7. Cross-Site Scripting (XSS)

- **Nedir:** Kötü niyetli kodların, genellikle JavaScript'in, başka kullanıcıların tarayıcılarında çalıştırılması.
- **Neden Kaynaklanır:** Kullanıcı girdilerinin yeterince doğrulanmaması.
- **Nasıl Önlenir:** Girdilerin temizlenmesi ve güvenlik başlıklarının (security headers) kullanılması.

8. Insecure Deserialization

- **Nedir:** Serileştirilmiş (serialized) verilerin güvensiz bir şekilde işlenmesi sonucu ortaya çıkan güvenlik açıkları.
- **Neden Kaynaklanır:** Deserialization işlemi sırasında veri manipülasyonunun kontrol edilmemesi.
- **Nasıl Önlenir:** Serileştirilmiş verileri doğrulamak ve dijital imza kullanmak.

9. Using Components with Known Vulnerabilities

- **Nedir:** Zafiyeti bilinen yazılım bileşenlerinin (kütüphaneler, framework'ler) kullanılması.
- **Neden Kaynaklanır:** Güncellemelerin ve güvenlik yamalarının zamanında uygulanmaması.
- **Nasıl Önlenir:** Kullanılan bileşenleri düzenli olarak güncellemek ve güvenlik açıklarını takip etmek.

10. Insufficient Logging & Monitoring

- **Nedir:** Sistemlerde yeterli kayıt tutma ve izleme yapılmaması, saldırıların tespit edilmesini zorlaştırır.
- **Neden Kaynaklanır:** Yetersiz loglama ve izleme yapılandırmaları.
- **Nasıl Önlenir:** Güvenlik olaylarını izlemek için yeterli loglama ve alarm sistemleri kurmak.